Jeu Programmation avec Unity et C#

Guide complet du débutant

Casey Hardman

Apress®

Machine Translated by Google

Programmation de jeux avec Unity et C#

Guide complet du débutant

Casey Hardman

Apress®

Machine Translated by Google

Programmation de jeux avec Unity et C#

Casey Hardman
West Palm Beach, Floride, États-Unis

ISBN-13 (pbk) : 978-1-4842-5655-8 https://doi.org/10.1007/978-1-4842-5656-5 ISBN-13 (électronique) : 978-1-4842-5656-5

Droits d'auteur © 2020 par Casey Hardman

Cet ouvrage est protégé par le droit d'auteur. Tous droits sont réservés par l'éditeur, que ce soit pour l'ensemble ou une partie du contenu, et notamment les droits de traduction, de reproduction, de réutilisation des illustrations, de récitation, de diffusion, de reproduction sur microfilm ou par tout autre moyen physique, de transmission, de stockage et de récupération de l'information, d'adaptation électronique, de logiciel informatique ou par toute autre méthode similaire ou différente, connue ou à venir.

Des noms, logos et images de marques déposées peuvent figurer dans cet ouvrage. Au lieu d'utiliser systématiquement le symbole de marque déposée, nous les utilisons uniquement à des fins éditoriales et dans l'intérêt du propriétaire de la marque, sans aucune intention de porter atteinte à ses droits.

L'utilisation dans cette publication de noms commerciaux, de marques de commerce, de marques de service et de termes similaires, même s'ils ne sont pas identifiés comme tels, ne doit pas être interprétée comme une prise de position quant à savoir s'ils sont ou non soumis à des droits de propriété.

Bien que les conseils et informations contenus dans cet ouvrage soient considérés comme exacts et véridiques à la date de publication, ni les auteurs, ni les éditeurs, ni l'éditeur ne peuvent être tenus responsables des erreurs ou omissions éventuelles. L'éditeur n'offre aucune garantie, expresse ou implicite, quant au contenu de ce livre.

Directeur général, Apress Media LLC : Welmoed Spahr
Responsable des acquisitions : Spandana Chatterjee
Éditeur en chef du développement : James Markham
Rédactrice en chef : Divya Modi

Couverture conçue par eStudioCalamar

Image de couverture conçue par Pixabay

Distribué aux libraires du monde entier par Springer Science+Business Media New York, 1 New York Plaza, New York, NY 10004. Téléphone : 1-800-SPRINGER, télécopie : (201) 348-4505, courriel : orders-ny@springer-sbm.com, ou consultez le site www.springeronline.com. Apress Media, LLC est une SARL californienne dont l'unique associé (propriétaire) est Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc est une société du Delaware.
corporation.

Pour toute information concernant les traductions, veuillez envoyer un courriel à rights@apress.com ou consulter le site http://www.apress.com/ droits et autorisations.

Les titres Apress peuvent être achetés en gros pour un usage académique, professionnel ou promotionnel. Des versions numériques et des licences sont également disponibles pour la plupart des titres. Pour plus d'informations, consultez notre page Web « Ventes groupées de livres imprimés et numériques » à l'adresse http://www.apress.com/bulk-sales.

Tout code source ou autre document complémentaire mentionné par l'auteur dans cet ouvrage est accessible aux lecteurs sur GitHub via la page produit du livre, disponible à l'adresse www.apress.com/978-1-4842-5655-8. Pour plus d'informations, veuillez consulter http://www.apress.com/ source-code.

Imprimé sur du papier sans acide

Machine Translated by Google

Table des matières

À propos de l'auteur xv

À propos du réviseur technique xvii

Introduction

Chapitre 1 : Installation et configuration 1

Installation d'Unity 1

Installation de notre éditeur de code 4

Création d'un projet 7

Résumé 9

Chapitre 2 : Notions de base d'Unity 11

Windows 11

Fenêtre du projet 12

Fenêtre de scène 13

Fenêtre de hiérarchie 14

Inspecteur Fenêtre 14

Composants 14

Ajout d'objets de jeu 17

Résumé 20

Chapitre 3 : Manipuler la scène 21

Outils de transformation 21

Positions et axes 23

Création d'un plancher 24

Échelles et unités de mesure 24

Résumé 26

Machine Translated by Google

Table des matières

Chapitre 4 : Les parents et leurs enfants 29

Objets de jeu enfant 29

Coordonnées mondiales vs. locales 32

Un bâtiment simple 33

Points pivots 36

Résumé 39

Chapitre 5 : Maisons préfabriquées

Fabrication et mise en place de préfabriqués 41

Édition des préfabriqués 42

Valeurs de remplacement 44

Préfabriqués imbriqués 48

Variantes préfabriquées 49

Résumé 50

Chapitre 6 : Introduction à la programmation 53

Langages de programmation et syntaxe 53

Quel code correspond à 54 ?

Typage fort vs. faible 55

Extensions de type de fichier 57

Scripts 57

Résumé 59

Chapitre 7 : Blocs de code et méthodes

Déclarations et points-virgules 61

Blocs de code 61

Commentaires 63

Méthodes 64

Méthodes d'appel 67

Types de données de base 69

Valeurs de retour avec les méthodes 69

Méthodes de déclaration 70

Machine Translated by Google

Opérateurs 73

Résumé 75

Chapitre 8 : Conditions 77

Le bloc « si » 77

Surcharges 79

Énumérations 80

Le bloc « else » 81

Le bloc « else if » 82

Opérateurs pour les conditions 83

Opérateurs d'égalité 83

Supérieur à et inférieur à 85

Ou 85

Et 86

Résumé 86

Chapitre 9 : Travailler avec des objets

Classes 87

Variables 89

Accès aux membres du groupe 90

Méthodes d'instance 93

Déclaration des constructeurs 95

Utilisation du constructeur 98

Membres statiques 99

Résumé 101

Chapitre 10 : Utilisation des scripts

Utilisations et espaces de noms 104

Classe de script 105

Rotation d'une transformation 107

Images et secondes 109

Attributs 110

Résumé 112

Machine Translated by Google

Table des matières

Chapitre 11 : L'héritage

Héritage en action : Objets RPG 113

Déclaration de nos classes 115

Chaînage de constructeurs 118

Sous-types et conversion de type 122

Types de valeurs numériques 124

Vérification de type 126

Méthodes virtuelles 127

Résumé 128

Chapitre 12 : Débogage 129

Configuration du débogueur 129

Points de rupture 131

Utilisation de la documentation Unity 135

Résumé 137

Partie I : Parcours d'obstacles 139

Chapitre 13 : Conception et plan du parcours d'obstacles 141

Aperçu du gameplay 142

Aperçu technique 143

Commandes du joueur 143

Mort et réapparition 145

Niveaux 145

Sélection du niveau 146

Obstacles 146

Configuration du projet 146

Résumé 148

Chapitre 14 : Déplacement du joueur 149

Configuration du joueur 150

Matériaux et couleurs 152

Déclaration de nos variables 155

Machine Translated by Google

Propriétés 158

Suivi de la vitesse 159

Application du mouvement 166

Résumé 170

Chapitre 15 : Mort et renaissance 173

Activation et désactivation 174

Méthode de mort 177

Méthode de réapparition 178

Résumé 180

Chapitre 16 : Risques fondamentaux 181

Détection de collision 181

Script de danger 188

Script de projectile 190

Scénario de tournage 194

Résumé 197

Chapitre 17 : Murs et objectifs

Murs 199

Objectifs 201

Paramètres de création des scènes 204

Résumé 207

Chapitre 18 : Dangers liés aux patrouilles 209

Ressemblant à un point de patrouille 209

Tableaux 210

Mise en place des points de patrouille 212

Détection des points de patrouille 216

La boucle « for » 218

Points de contrôle de tri 221

Déplacement du Patroller 226

Résumé 229

Machine Translated by Google

Table des matières

Chapitre 19 : Dangers errants 233

Régions de voyage 233

Extension d'éditeur de base 236

Scripts de l'éditeur 236

Inspecteurs des douanes 237

Accès à la cible de l'inspecteur 237

Dessiner sur scène 238

Configuration du vagabond 240

Scénario du Vagabond 241

Gestion de l'État

Réaction à l'État

Résumé 247

Chapitre 20 : Époustouflant 249

Variables dynamiques 249

Méthode Dashing 251

Dernières retouches 254

Temps de recharge du dash 255

Résumé 257

Chapitre 21 : Conception des niveaux 259

Préfabriqués et variantes 259

Création de niveaux 261

Ajout de murs 263

Caméra de vue au niveau du sol 264

Résumé 264

Chapitre 22 : Menus et interface utilisateur 267

Déroulement de la scène 267

Script de sélection de niveau 269

Résumé 276

Machine Translated by Google

Chapitre 23 : Menu Pause en jeu 279

Temps de congélation 279

Résumé 284

Chapitre 24 : Pièges à pointes

Conception du piège 285

Levée et abaissement 289

Écriture du scénario 290

Ajout de collisions 295

Résumé 297

Chapitre 25 : Conclusion du parcours d'obstacles 299

Construction du projet

Paramètres du lecteur 301

Récapitulatif 303

Fonctionnalités supplémentaires 304

Résumé 306

Partie II : Défense de tour 307

Chapitre 26 : Conception et plan d'un système de défense de tour 309

Aperçu du gameplay 309

Aperçu technique 311

Configuration du projet 313

Résumé 313

Chapitre 27 : Mouvements de caméra 315

Configuration 315

Déplacement par les touches fléchées 318

Application du mouvement 320

Glissement de la souris 322

Zoom 324

Résumé 325

Machine Translated by Google

Table des matières

Chapitre 28 : Ennemis, tours et projectiles 327

Couches et physique 327

Ennemis de base 329

Projectiles 333

Ciblage 339

Tours 348

Tours Arrow 351

Résumé 359

Chapitre 29 : Mode construction 361

Principes de base de l'interface utilisateur 362

La transformation rectangulaire 366

Création de notre interface utilisateur 368

Événements 373

Configuration 374

Logique du mode construction 379

Le dictionnaire 385

Méthodes d'événement OnClick 388

Résumé 395

Chapitre 30 : Mode Jeu 397

Points d'apparition et de fuite 397

Verrouillage du bouton Lecture 399

Configuration de la recherche de chemin 401

Trouver un chemin 404

Configuration du mode de jeu 409

Apparition des ennemis 414

Mouvement ennemi 417

Résumé 423

Machine Translated by Google

Chapitre 31 : Autres types de tours 425
Projectiles à trajectoire parabolique 425
Tour du Canon 431
Plaques chauffantes 435
Barricades 437
Résumé 437
Chapitre 32 : Conclusion de Tower Defense 439
Héritage 439
UI 441
Raycasting 441
Recherche de chemin 442
Fonctionnalités supplémentaires 442
Barres énergétiques 442
Types d'armure et de dégâts 443
Tracé plus complexe 443
Indicateurs de plage 444
Amélioration des tours 445
Résumé 445
Partie III : Terrain de jeu de la physique 447
Chapitre 33 : Conception et aménagement du terrain de jeu de physique 449
Présentation des fonctionnalités 449
Caméra 449
Mouvements des joueurs 450
Pousser et tirer 450
Plateformes mobiles 451
Balançoires 451
Champs de force et plateformes de saut 451
Configuration du projet 451
Résumé 452

Machine Translated by Google

Table des matières

Chapitre 34 : Caméra à visée par souris 453

Configuration du joueur 453

Comment ça marche 454

Configuration du script 456

Raccourcis clavier 463

Entrée souris 464

Mode Première Personne 469

Mode à la troisième personne 470

Test 474

Résumé 475

Chapitre 35 : Mouvements 3D avancés 477

Comment ça marche 477

Script du joueur 480

Vitesse de déplacement 486

Application du mouvement 490

Perte de vitesse 493

Gravité et sauts 494

Résumé 496

Chapitre 36 : Saut mural 497

Variables 497

Détection des murs 499

Exécution du saut 502

Résumé 505

Chapitre 37 : Tirer et pousser 507

Configuration du script 507

Mise à jour corrigée 511

Détection de cibles 513

Tirer et pousser 515

Machine Translated by Google

Dessin du curseur 517

Résumé 518

Chapitre 38 : Plateformes mobiles 519

Mise en place de la scène 520

Mouvement de la plateforme 522

Jeu de plateforme pour joueurs 529

Résumé 532

Chapitre 39 : Articulations et balancements

Installation de la balançoire 533

Raccordement des articulations 540

Touches finales 542

Résumé 543

Chapitre 40 : Champs de force et plateformes de saut 545

Configuration du script 545

Configuration du champ de force 547

Ajout de vélocité au joueur 548

Application des forces 549

Résumé 552

Chapitre 41 : Conclusion 553

Récapitulatif du terrain de jeu de physique 553

Apprentissage complémentaire pour l'unité 555

La boutique d'actifs 555

Terrains 555

Coroutines 556

Ordre d'exécution du script 556

Formation complémentaire en C# 557

Délégués 557

Commentaires sur la documentation 558

Exceptions 560

Machine Translated by Google

Table des matières

C# avancé 562

Surcharge de l'opérateur 562

Conversions 562

Types génériques 563

Structures 563

Résumé 563

Index 565

Machine Translated by Google

À propos de l'auteur

Casey Hardman est un développeur de jeux vidéo amateur, inspiré par le potentiel d'immersion et d'interactivité des jeux. Il se spécialise dans le moteur de jeu Unity. Passionné de jeux vidéo depuis son enfance, il s'est lancé, au début de son adolescence, dans le monde de la conception et de la programmation de jeux.

Il est autodidacte et a appris grâce à divers projets personnels, des plus modestes aux plus ambitieux. Il contribue régulièrement à différentes plateformes de développement de jeux en ligne et passe beaucoup trop de temps devant son ordinateur.

Machine Translated by Google

À propos du réviseur technique

Robert Lair développe des logiciels professionnellement depuis plus de 25 ans et a occupé quasiment tous les postes du cycle de vie du développement logiciel (SDLC). Il a été président-directeur général, vice-président du développement produit, directeur technique, architecte logiciel, développeur et Scrum Master/ chef de produit. Il considère le développement de logiciels de qualité comme un art et se passionne pour les architectures logicielles rigoureuses, les processus de développement efficaces, la documentation, la réduction de la dette technique, l'organisation et la productivité. Il se spécialise dans la création d'applications Unity, web et mobiles axées sur la gamification. Pour en savoir plus sur Robert, consultez son site web : www.robertlair.com.

Machine Translated by Google

Introduction

Bienvenue dans votre aventure de programmation de jeux avec Unity. Ce livre est conçu pour vous apprendre à programmer des jeux vidéo de A à Z, tout en vous proposant de nombreux exercices pratiques. Il ne s'agit pas de réaliser des projets ambitieux ni de créer des graphismes sophistiqués. Nous apprenons à programmer et à utiliser le moteur Unity. Une fois ces notions essentielles acquises, vous pourrez approfondir vos connaissances et créer des jeux toujours plus complexes et impressionnants.

Tous les logiciels utilisés sont multiplateformes. Ce livre privilégiera la terminologie et les exemples Windows, mais vous pourrez facilement l'adapter à d'autres systèmes d'exploitation majeurs, comme macOS ou Linux.

Concernant la configuration système requise, tout ordinateur récent acheté au cours des six dernières années devrait pouvoir exécuter sans problème le logiciel que nous utiliserons. Comme nous n'avons pas recours à des graphismes haut de gamme ni à des algorithmes complexes, les exemples de projets que nous développerons devraient fonctionner correctement sur la plupart des systèmes. En cas de doute, la configuration système officielle et la plus récente pour Unity est disponible sur le site officiel. site web ici :

https://unity3d.com/unity/system-requirements

Dans les chapitres 1 à 12, nous commencerons par une introduction aux concepts essentiels du moteur de jeu Unity et nous configurerons tous les outils nécessaires. Ensuite, nous aborderons les aspects pratiques de la programmation et nous commencerons à écrire du code.

Dans la suite de cet ouvrage, nous aborderons des projets de jeux individuels un par un, en créant des projets jouables que vous pourrez enrichir ultérieurement à votre guise. C'est là que vous acquerrez la majeure partie de votre expérience pratique. Nous implémenterons de véritables mécanismes de jeu, ce qui est ce que vous recherchez avant tout en tant que programmeur de jeux, n'est-ce pas ?

Le premier projet de jeu, « Parcours d'obstacles » (chapitres 13 à 25), sera un parcours d'obstacles en vue de dessus où le joueur déplacera son personnage avec les touches WASD ou les flèches directionnelles pour éviter divers dangers : patrouilles, ennemis errants, projectiles et pièges à pointes au sol. Nous nous familiariserons avec les déplacements et rotations de base, la création de niveaux, les concepts fondamentaux d'Unity comme les prefabs et le scripting, ainsi que la conception de l'interface utilisateur.

Machine Translated by Google

Introduction

Le projet de jeu 2, « Tower Defense » (chapitres 26 à 32), sera la base d'un jeu simple de « tower defense », où le joueur place des structures défensives sur le terrain de jeu.
Les ennemis se déplaceront d'un côté à l'autre du terrain, et les défenses du joueur tenteront de les repousser.
Nous explorerons les bases de la recherche de chemin (comment les ennemis contournent les obstacles) et approfondirons les concepts fondamentaux de la programmation.

Le projet de jeu n° 3, « Terrain de jeu physique » (chapitres 33 à 41), sera un terrain de jeu physique en 3D avec prise en charge des vues à la première et à la troisième personne pour un personnage joueur doté de mouvements plus complexes à la souris, de sauts, de sauts muraux et d'un système de gravité. Nous explorerons les possibilités du moteur physique d'Unity, de la détection d'objets par lancer de rayons à la configuration des articulations et des corps rigides.

Machine Translated by Google

CHAPITRE 1

Installation et configuration

L'installation d'un logiciel est plutôt simple : on télécharge un programme d'installation, on l'exécute, un menu (parfois appelé « assistant ») apparaît, on accepte les conditions d'utilisation, on choisit l'emplacement d'installation sur l'ordinateur, on propose parfois des options supplémentaires, et l'installation démarre. Facile, non ? Je ne vais donc pas détailler la procédure d'installation. Je vais simplement vous montrer ce qu'il faut installer.

Installation d'Unity

Unity publie régulièrement de nouvelles versions intégrant de nouvelles fonctionnalités, des correctifs et des améliorations mineures. C'est pourquoi ils ont récemment lancé Unity Hub.
C'est une petite application légère qui permet d'installer le moteur Unity, y compris ses anciennes versions. Elle permet également de gérer les anciennes versions du moteur déjà installées sur votre ordinateur et de visualiser tous vos projets Unity depuis une interface unique.

Il est parfois utile de conserver une ancienne version d'Unity même après la mise à jour. à la dernière version. Il peut être judicieux de travailler sur un projet plus ancien avec la même version que celle utilisée initialement, au cas où certaines nouvelles fonctionnalités ou modifications ne seraient pas compatibles avec votre ancien projet ; les choses évoluent, et il arrive que les nouveautés rendent les anciennes incompatibles. Parfois, l'ancienne version est simplement remaniée et n'est plus compatible avec un moteur plus récent. Dans ce cas, vous pouvez décider de conserver l'ancienne version jusqu'à la fin de votre projet, afin d'éviter de perdre du temps à adapter vos méthodes de travail à la « nouvelle méthode ».

Nous allons donc commencer par installer Unity Hub, puis installer le moteur Unity lui-même via le Hub. Pour télécharger le Hub, rendez-vous sur ce lien dans votre navigateur web :

https://unity3d.com/get-unity/download

Cliquez sur le bouton intitulé « Télécharger Unity Hub », comme indiqué dans la figure 1-1.

© Casey Hardman 2020
C. Hardman, Programmation de jeux avec Unity et C#, https://doi.org/10.1007/978-1-4842-5656-5_1

Machine Translated by Google

Figure 1-1. Bouton de téléchargement Unity Hub

Le programme d'installation du Hub va se télécharger. Exécutez-le et suivez les instructions.

Une fois le Hub installé, lancez-le. Unity pourrait vous empêcher d'aller très loin.
Il vous sera demandé d'accepter une licence, ce qui peut impliquer la création d'un compte Unity. Cette configuration est rapide et ponctuelle ; une fois terminée, votre compte restera actif. Cela ne vous posera plus de problème. Toutefois, si la création d'un compte vous est proposée, n'oubliez pas votre nom d'utilisateur et votre mot de passe !

Pour comprendre ce qu'est une « licence », sachez qu'Unity proposait auparavant une version gratuite et une version Pro. Certaines fonctionnalités étaient limitées dans la version gratuite, et il fallait acheter la version Pro pour y accéder — notamment les fonctionnalités avancées, en particulier les effets et éclairages 3D sophistiqués. Par la suite, la quasi-totalité des fonctionnalités a été rendue accessible à la version gratuite, tandis que les versions payantes offraient principalement des avantages divers, tels qu'une assistance renforcée, des ressources supplémentaires et des outils de collaboration d'équipe. Désormais, trois types de licences sont proposés pour utiliser le moteur : Personnelle, Plus et Pro.

Tant que vous ou l'entreprise que vous représentez avez réalisé un chiffre d'affaires brut inférieur à 100 000 $ l'année précédente (c'est notre cas), vous pouvez utiliser la licence Unity Personnelle, qui est gratuite. Si votre carrière de développeur de jeux décolle et que vous commencez à gagner de l'argent, vous devrez passer à Unity Plus (25 $/mois) ou Unity Pro (125 $/mois).
(mois) pour éviter d'enfreindre la licence. Mais n'allons pas trop vite en besogne : pour l'instant, nous ne sommes que des amateurs.

Il se peut également qu'on vous demande de remplir un petit questionnaire. Il s'agit simplement de quelques questions concernant votre utilisation prévue du moteur, vos centres d'intérêt et d'autres informations de base.

Une fois que vous avez une licence et un compte, vous devriez voir un onglet « Installations » sur le côté gauche du Hub (voir Figure 1-2).

Machine Translated by Google

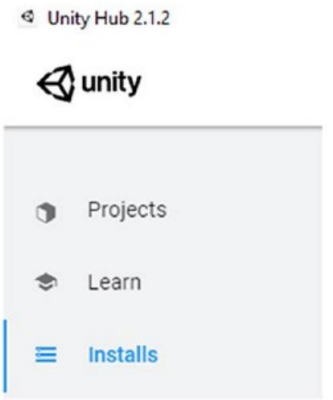

Figure 1-2. Coin supérieur gauche du Unity Hub, avec l'onglet Installations sélectionné

Vous pouvez consulter ici toutes les versions du moteur Unity installées sur votre ordinateur. Vous pouvez également installer de nouvelles versions ; toutefois, l'installation de nombreuses versions peut rapidement saturer l'espace disque. Il est donc conseillé de désinstaller les anciennes versions pour éviter cela. Dans l'onglet « Installations », cliquez sur le bouton « Ajouter » en haut à droite.
Une fenêtre contextuelle affichera une liste de versions disponibles. La version en haut de la liste sera la dernière version stable. Cliquez dessus pour la sélectionner, puis cliquez sur « Suivant » en bas à droite.

Il vous sera demandé de sélectionner ce que Unity Hub appelle des « modules » à installer avec le moteur. Ce sont des fonctionnalités supplémentaires qui s'ajoutent à l'installation et qui occupent de l'espace disque supplémentaire si vous choisissez de les installer. Vous pouvez toujours ajouter des modules ultérieurement, une fois l'installation terminée, si vous le souhaitez. Les modules de support de compilation sont les plus importants : ils vous permettent de compiler un projet de jeu Unity pour différents systèmes d'exploitation, environnements et configurations matérielles.

« Construire » un projet de jeu est le processus qui consiste à le transformer d'un projet Unity en un jeu jouable. uniquement via le moteur Unity, vers une application réelle utilisée pour jouer au jeu.

Les principales plateformes sur lesquelles les projets Unity peuvent être construits sont :

- PC, compatible avec Windows, Mac et Linux
- Android
- Apple iOS
- WebGL (joué dans un navigateur web)
- Xbox One
- PS4

Machine Translated by Google

Comme je l'ai dit, si vous devez développer pour ces plateformes ultérieurement (nous n'entrerons pas dans les détails), (les dans ce livre), puis vous pouvez toujours les installer via le Hub.

Vous pouvez également choisir d'installer la documentation Unity localement, sous forme de module. La documentation peut s'avérer extrêmement utile. Elle est également disponible en ligne, et j'ai moi-même tendance à privilégier les ressources en ligne, mais si vous prévoyez d'utiliser Unity hors ligne, il peut être judicieux d'installer la documentation afin d'y avoir accès sans connexion Internet.

Pour l'instant, cochez simplement la case « Prise en charge de la compilation pour votre système d'exploitation » et, si vous le souhaitez, celle pour la documentation, puis décochez toutes les autres options. Cliquez ensuite sur « Terminé » pour lancer l'installation. La version en cours d'installation s'affichera dans une case au centre de la fenêtre, et une barre de progression indiquera l'avancement de l'installation.

Une fois l'installation terminée, la barre disparaîtra et vous pourrez créer un projet Unity avec cette version du moteur. Nous allons le faire dans un instant et lancer le moteur pour la première fois. Autre avantage du Hub : il lancera automatiquement la version appropriée de l'éditeur Unity lorsque vous ouvrirez un projet (à condition qu'elle soit installée sur votre ordinateur).

Installation de notre éditeur de code

On n'écrit pas du code avec le même type de logiciel qu'un livre ou un CV. Les éditeurs de code sont des éditeurs de texte spécialement conçus pour l'écriture de code. Ils proposent une coloration syntaxique pour les mots et les symboles, savent formater le code et offrent souvent une multitude de fonctionnalités qui facilitent et accélèrent l'écriture et la manipulation du code.

Notre éditeur de code de prédilection est Microsoft Visual Studio Code. Attention à ne pas le confondre avec Microsoft Visual Studio. Ce sont deux produits similaires (et portant un nom similaire) de la même entreprise, tous deux gratuits, et vous pouvez utiliser l'un ou l'autre pour réaliser votre projet. Ils permettent tous deux de modifier du code C# et de s'intégrer à Unity.

Visual Studio Code est conçu pour être multiplateforme et léger dès son installation. mais hautement extensible. Son interface utilisateur est minimaliste et la plupart de ses fonctionnalités sont activées par l'installation d'extensions permettant d'ajouter des fonctionnalités supplémentaires, ce qui se fait directement dans le logiciel.

Visual Studio est compatible avec Windows et macOS, mais pas avec Linux. Il offre davantage de fonctionnalités prêtes à l'emploi. C'est un outil très puissant aux multiples applications, notamment la collaboration en équipe et d'autres fonctionnalités avancées. Il est généralement plus gourmand en ressources : plus riche en fonctionnalités, il consomme davantage de mémoire et est donc légèrement plus lent.

Machine Translated by Google

Nous avons choisi Code car je trouve qu'il convient mieux aux débutants, car il ne les « gêne pas » autant, et la plupart des fonctionnalités avancées de Visual Studio ne correspondent pas vraiment à notre flux de travail.

Pour télécharger Code, rendez-vous sur ce lien dans votre navigateur web préféré :

https://code.visualstudio.com/download

Vous pouvez ensuite sélectionner le bouton correspondant à votre système d'exploitation (Windows, Linux ou Mac) pour télécharger le logiciel. Le téléchargement de l'installateur devrait alors démarrer. Une fois terminé, exécutez le fichier et suivez les instructions.

Une fois Visual Studio Code lancé pour la première fois, vous verrez un message de bienvenue. Cette page centralise divers liens et ressources. De nombreux programmeurs sont exigeants quant au thème de couleurs de leur éditeur de code. Personnellement, je préfère les thèmes sombres. D'autres préfèrent les thèmes clairs ; à vous de choisir. Vous pouvez facilement le modifier directement depuis la page d'accueil. Cliquez sur le bouton « Thème de couleur » pour faire apparaître une liste de thèmes de couleurs standard, vous permettant de choisir le thème que vous préférez (voir Figure 1-3).

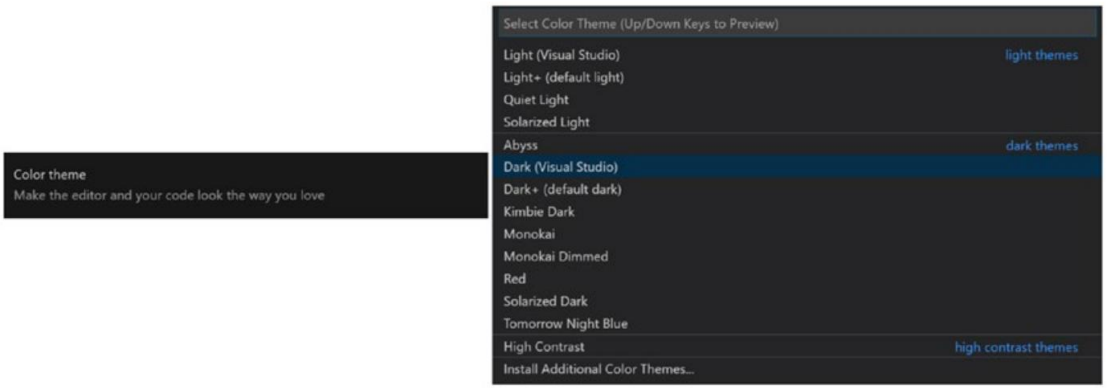

Figure 1-3. Page d'accueil : bouton « Thème de couleurs » (à gauche) et fenêtre contextuelle correspondante (à droite).

Une fois vos couleurs à votre goût (vous pourrez les modifier ultérieurement), fermons la page d'accueil. Chaque fichier ou page ouvert dans Code possède un onglet en haut à gauche (voir Figure 1-4), qu'il s'agisse d'un fichier de code modifiable ou d'une page statique comme la page d'accueil. Si plusieurs fichiers sont ouverts, vous pouvez facilement passer de l'un à l'autre en cliquant sur l'onglet. Pour l'instant, seule la page d'accueil est ouverte. Fermons-la pour repartir de zéro. Vous pouvez le faire en cliquant sur le bouton « X » de l'onglet ou en utilisant le raccourci clavier Ctrl+W.

Machine Translated by Google

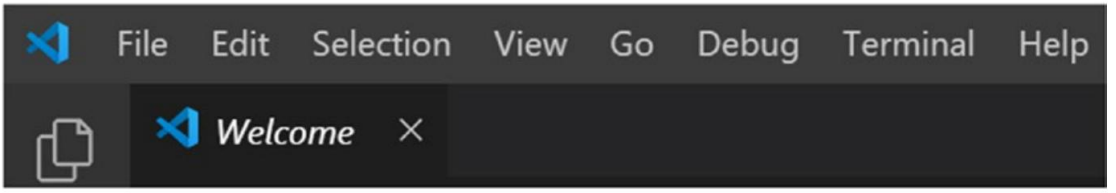

Figure 1-4. Coin supérieur gauche de Code, où sont affichés tous les onglets de la page. Seul l'onglet de la page d'accueil est ouvert ici.

Une fois fermé, il ne se passera plus grand-chose – juste un grand espace vide, comme le montre l'image. Figure 1-5.

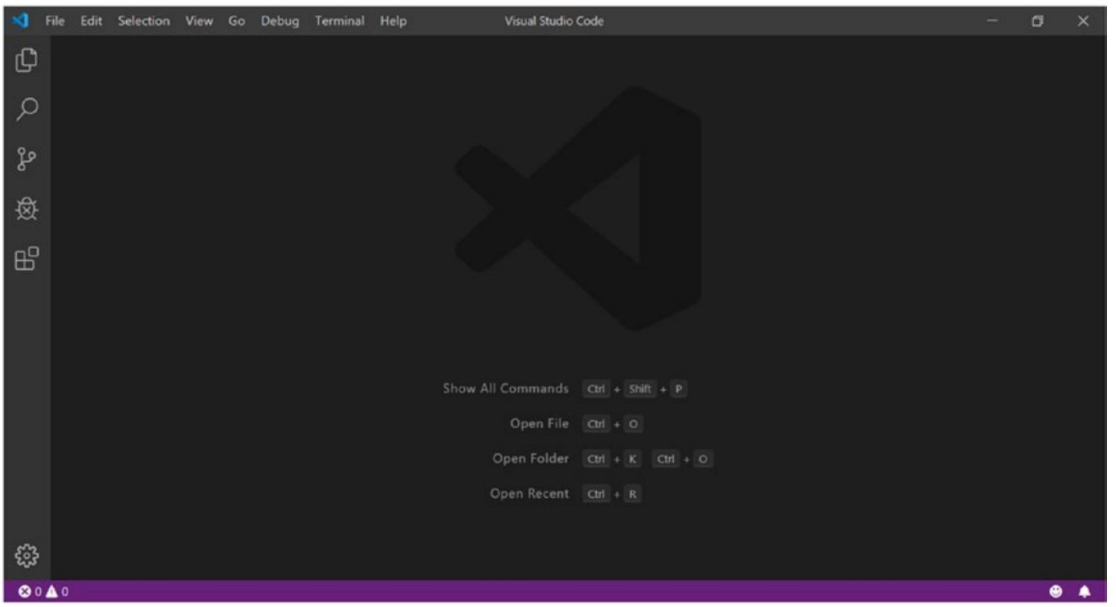

Figure 1-5. Vue de Visual Studio Code sans aucun fichier ouvert

Vous pouvez désormais installer ce que Code appelle des « extensions » pour ajouter des fonctionnalités supplémentaires à L'éditeur. Les extensions sont gérées et installées via un bouton situé dans la barre latérale gauche. La barre latérale gauche peut prendre différentes formes, selon le dernier bouton cliqué. Survolez les boutons avec la souris pour afficher leur signification, puis cliquez sur le bouton Extensions (en bas). Vous pouvez également utiliser le raccourci Ctrl+Maj+X pour l'afficher. La barre latérale apparaîtra alors à gauche de l'écran. Un nouvel appui sur le bouton la rétractera.

Machine Translated by Google

Dans la barre latérale Extensions, vous pouvez rechercher des extensions à l'aide de la barre de recherche située en haut. Cliquez dans cette barre et saisissez « C# ». Un résultat intitulé simplement « C# » devrait s'afficher, accompagné de la description « C# pour Visual Studio Code ». Vous remarquerez également que l'éditeur de l'extension est indiqué sous la description : « Microsoft », qui est la société à l'origine de Visual Studio Code et du langage C#.

Cliquez sur cette extension ; un nouvel onglet s'ouvrira avec des informations détaillées. Sous la description principale, en haut de la page, vous trouverez un bouton d'installation. Cliquez dessus pour lancer l'installation. Si des fenêtres contextuelles vous proposent d'installer d'autres extensions, autorisez leur installation.

Ensuite, nous installerons l'extension de débogage pour Unity. Cela nous permettra de connecter notre éditeur de code à Unity afin de définir des points d'arrêt. Un point d'arrêt est un endroit du code qui, lorsqu'il est atteint pendant l'exécution du programme, le fige. Pendant ce temps, nous pouvons examiner les données du programme et reprendre l'exécution à tout moment. C'est une fonctionnalité très pratique que nous utiliserons plus tard.

Le nom exact de l'extension est « Debugger for Unity », publiée par Unity Technologies. Vous pouvez la trouver de la même manière : en saisissant son nom dans la barre de recherche située en haut de la barre latérale Extensions.

Une fois ces extensions installées, nous pouvons fermer Visual Studio Code. Nous serons Nous l'utiliserons davantage dans la partie 2. Pour l'instant, nous nous concentrons sur l'éditeur Unity lui-même.

Création d'un projet

Nous pouvons maintenant utiliser Unity Hub pour créer notre premier projet et ainsi disposer d'un environnement pour expérimenter et apprendre. Dans Unity Hub, cliquez sur l'onglet Projets à gauche, puis sur le bouton bleu « Nouveau » en haut à droite.

Une boîte de dialogue apparaîtra (voir Figure 1-6), vous permettant de sélectionner un modèle sur lequel baser le projet.

Machine Translated by Google

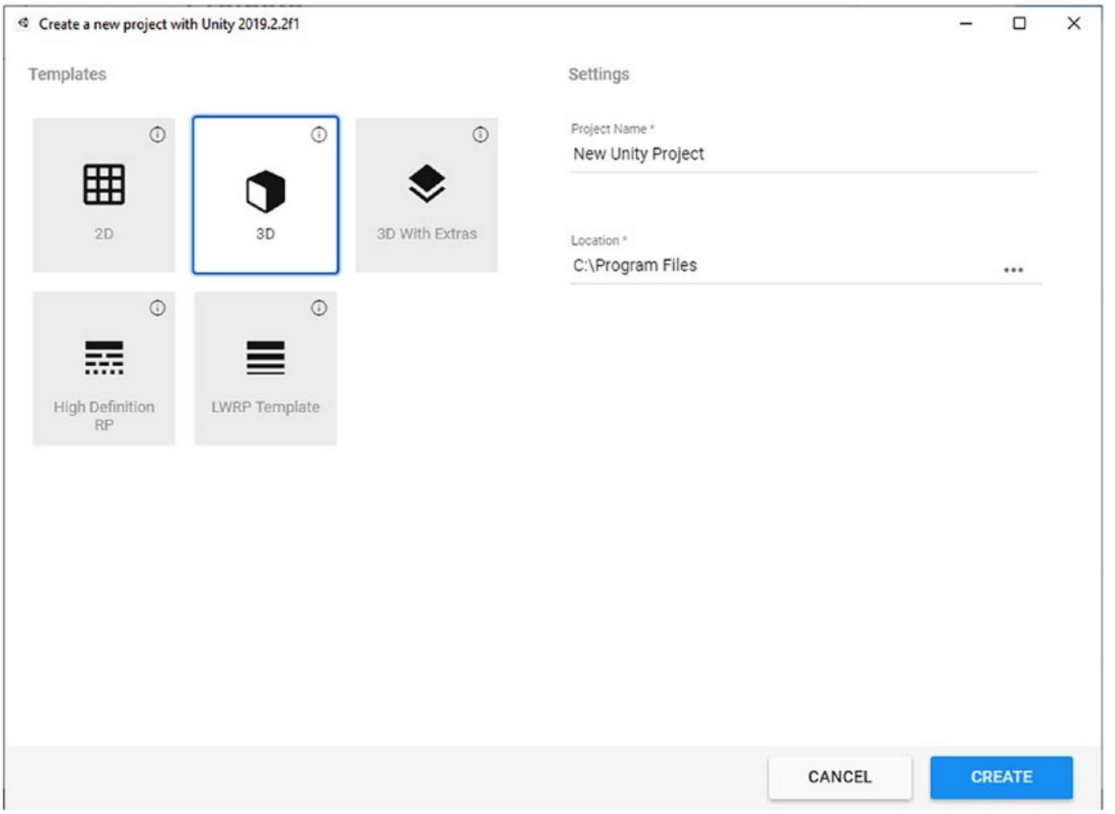

Figure 1-6. Boîte de dialogue pour la création d'un nouveau projet Unity

Le modèle n'est qu'un point de départ pour votre projet. Les deux premiers, intitulés « 2D » et « 3D », sont les plus simples et les plus basiques. Ce sont des modèles vierges, l'un configuré pour la 2D et l'autre pour la 3D.

Nous allons commencer par un projet 3D vierge ; sélectionnez donc le modèle 3D en cliquant dessus. Il contiendra Une bordure bleue apparaît autour de l'élément sélectionné.

À droite des options du modèle, vous trouverez un champ pour le nom du projet (vous pourrez le modifier ultérieurement) et un autre pour le répertoire (chemin d'accès) sur votre ordinateur où enregistrer les fichiers du projet. Nous nommerons notre projet « ExampleProject » ; notez que nous n'utilisons pas d'espace entre les deux mots, car les chemins d'accès ne les acceptent pas toujours.

Vous pouvez choisir l'emplacement d'enregistrement de votre projet. Quel que soit le chemin d'accès sélectionné, un dossier portant le nom du projet sera créé à l'intérieur. Ce dossier constitue le répertoire racine de votre projet et contient tous ses fichiers et ressources.

Machine Translated by Google

Une fois le chemin d'accès souhaité sélectionné, cliquez sur le bouton bleu « Créer » en bas à droite et patientez pendant que Unity crée les fichiers de base du projet. Une fois la création terminée, l'éditeur s'ouvrira avec votre nouveau projet, prêt à être modifié.

Résumé

Voici un récapitulatif de ce que nous avons appris dans ce chapitre :

- Le programme Unity Hub permettra de télécharger les nouvelles versions de l'éditeur Unity, de désinstaller les anciennes versions dont vous n'avez plus besoin, de créer de nouveaux projets et d'ouvrir des projets existants.
- Ouvrir un projet dans Unity Hub lancera l'éditeur Unity, qui est l'endroit où nous utiliserons le moteur pour développer notre jeu.
- Un projet de jeu Unity est stocké sur votre ordinateur, avec tous les fichiers associés – y compris les éléments que nous créons nous-mêmes comme les graphismes et le code – stockés dans un « répertoire racine » portant le nom du projet.

Notre code sera écrit avec Visual Studio Code, un éditeur de texte.

Conçu spécifiquement pour l'écriture de code, il offre des fonctionnalités utiles absentes des éditeurs de texte classiques, facilitant ainsi la mise en forme et la navigation dans le code.

Machine Translated by Google

CHAPITRE 2

Principes de base d'Unity

Maintenant qu'Unity est installé et que nous avons un nouveau projet, familiarisons-nous avec le moteur. Après tout, c'est l'interface utilisateur avec laquelle nous interagirons fréquemment lors du développement de nos jeux ; il est donc important de bien la connaître dès maintenant.

Windows

Unity est divisé en différentes fenêtres ayant chacune une fonction spécifique. Chaque fenêtre possède un petit onglet dans son coin supérieur gauche, où est inscrit son nom, c'est-à-dire son type.

Comme dans de nombreux logiciels actuels, chaque fenêtre constitue un élément distinct du programme, repositionnable, redimensionnable, voire supprimable. D'autres types de fenêtres, actuellement inutilisés, n'apparaissent pas à l'écran ; toutefois, leur ajout est très rapide si nécessaire.

Vous remarquerez que si vous cliquez avec le bouton gauche sur l'un de ces onglets de fenêtre, maintenez le bouton de la souris enfoncé. En cliquant sur le bouton droit et en faisant glisser la fenêtre, vous pouvez la déplacer à un autre endroit du programme. Vous pouvez ainsi diviser l'espace d'une fenêtre pour en recouvrir une partie avec une autre. Si vous souhaitez interagir avec une fenêtre, il vous suffit généralement de la déplacer et d'observer la réaction d'Unity.

Vous pouvez également ancrer des fenêtres côte à côte pour placer leurs onglets l'un à côté de l'autre (voir figure 2-1). L'onglet sur lequel vous avez cliqué en dernier sera sélectionné et occupera tout l'espace de la fenêtre.

Figure 2-1. Deux onglets de fenêtre ancrés côte à côte dans le même espace. La fenêtre Projet est active et occupe tout l'espace, mais il est possible de donner le focus à la fenêtre Console en cliquant sur son onglet.

© Casey Hardman 2020
C. Hardman, Programmation de jeux avec Unity et C#, https://doi.org/10.1007/978-1-4842-5656-5_2

Machine Translated by Google

Unity vous permet également d'enregistrer toutes vos fenêtres actuelles, ainsi que leurs tailles et positions , dans une disposition que vous pouvez nommer. Pour cela, utilisez le petit bouton déroulant situé en haut à droite de l'éditeur Unity, intitulé « Disposition ». Cliquez dessus pour afficher toutes les dispositions prédéfinies disponibles, comme illustré dans la figure 2-2.

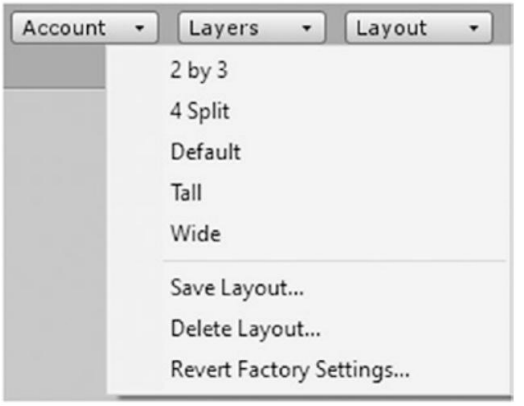

Figure 2-2. Le bouton « Disposition » situé dans le coin supérieur droit de l'éditeur Unity, après avoir cliqué dessus pour afficher ses options.

Par défaut, la disposition est configurée sur « Par défaut ». Essayez une autre disposition : Unity restructurera automatiquement toutes vos fenêtres. Vous pouvez également sélectionner « Enregistrer la disposition… » dans la liste déroulante pour sauvegarder votre disposition actuelle sous un nouveau nom et la conserver une fois que vous l'aurez configurée à votre convenance. Ainsi, si vous perdez une fenêtre par inadvertance ou si vous modifiez accidentellement votre configuration, vous pourrez la rétablir facilement en chargeant votre disposition via la liste déroulante.

Les schémas peuvent également s'avérer utiles pour aborder différents aspects du développement de jeux. Certaines activités sont plus faciles à réaliser avec une configuration de fenêtres différente et placée à un autre endroit. La possibilité d'enregistrer des configurations facilite le passage d'une activité à l'autre en quelques clics.

La configuration par défaut regroupe toutes les fenêtres les plus importantes. Passons-les en revue pour découvrir leur fonction.

Fenêtre de projet

La fenêtre Projet affiche tous nos éléments. Dans le jargon du développement de jeux vidéo, un élément désigne une ressource que nous pouvons utiliser dans notre jeu : graphismes, effets sonores, musique, fichiers de code, niveaux, etc.